

Link Lists, STL `vector<short>`, and the STL `sort()` algorithm

Part I:

- (0) Download the `LinkedList_STL_sortStarter.cpp` file from onCourse. Create a **link list** of N short integers in the range $[1..5000]$ and **print the values** in your link list separated by commas.

Call me over when this is working.

Part II:

- (1) **Move only the even integers** from your link list **into an STL vector**. **Print the values** in your vector.

A sample output of Part I and Part II (using only $N=5$):

LINK LIST: 1808, 3931, 250, 3659, 74, END.

VECTOR of EVENS: 1808, 250, 74, END.

Call me over when this is working.

Part III:

- (2) Run experiments using the `sort()` from the STL `<algorithm>` library. Using $N=20^8$ (200 million) at the very start of Part I (thus, we expect on average about 1M even integers to be in your vector), you should report (on each row in the Table) only the average runtime over five (5) different trials. Note you must take care to set up each (row) trial (preprocessing the data *before* sorting), for example, for “nearly sorted” data, you might sort the data at the start and then switch just a few values prior to the sorting test.

Try to get at least the initial row done and submitted.

Arrangement of (starting) data $N \approx 10^8$	<code>std::sort</code> (sec) average of 5 trials	<code>std::stable_sort</code> (sec) average of 5 trials
random values		
reversed at start		
nearly sorted		
mostly the same values (just a few unique values)		

- (3) Create an **Excel** table (Table 1) and graph (Figure 1) of your experimental data.

- (4) Submit via onCourse a .zipped folder of

- (i) your source code (only the source, no Project) and
- (ii) an Excel file containing your Excel table, and a graph